NOLTR 71-65

THE TRANSLATIONAL VELOCITY OF SURFACE SHIPS AND SUBMARINES: A COMPUTER PROGRAM

By
James R. Britt

7 MAY 1971

NOL

NAVAL ORDNANCE LABORATORY, WHITE OAK, SILVER SPRING, MARYLAND

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED

NOLTR 71-65

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Naval Ordnance Laboratory White Oak, Maryland 20910	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

The Translational Velocity of Surface Ships and Submarines: A Computer Program

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

5. AUTHOR(S) *(First name, middle initial, last name)*

James R. Britt

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
7 May 1971	iii + 30	3

8a. CONTRACT OR GRANT NO. DASA-NA 002-20 b. PROJECT NO. Prob. Assign. 106 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) NOLTR 71-65 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT

Approved for Public Release; Distribution Unlimited

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Defense Atomic Support Agency

13. ABSTRACT

ABSTRACT: This paper describes a computer program written in FORTRAN IV which calculates the peak or maximum translational velocity induced in both submerged and floating targets by an underwater explosion plane shock wave of arbitrary pulse shape. The targets, surface ships and submarines, are approximated by an infinitely long cylinder of a specified radius. The theory, which was developed for submarines, is described briefly and extended to floating targets.

Since the program was originally written to handle pulse shapes produced by reflections from the ocean bottom, it has the capability of using pulse shapes which may have a logarithmic singularity.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6801

Unclassified
Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Underwater Explosions						
Shock Damage to Surface Ships						
Shock Damage to Submarines						
Bottom Reflections						
Shock Waves						

THE TRANSLATIONAL VELOCITY OF SURFACE SHIPS AND SUBMARINES: A COMPUTER PROGRAM

By
James R. Britt

ABSTRACT: This paper describes a computer program written in FORTRAN IV which calculates the peak or maximum translational velocity induced in both submerged and floating targets by an underwater explosion plane shock wave of arbitrary pulse shape. The targets, surface ships and submarines, are approximated by an infinitely long cylinder of a specified radius. The theory, which was developed for submarines, is described briefly and extended to floating targets.

Since the program was originally written to handle pulse shapes produced by reflections from the ocean bottom, it has the capability of using pulse shapes which have a logarithmic singularity.

EXPLOSIONS RESEARCH DEPARTMENT
NAVAL ORDNANCE LABORATORY
WHITE OAK, SILVER SPRING, MARYLAND

NOLTR 71-65 7 May 1971

THE TRANSLATIONAL VELOCITY OF SURFACE SHIPS AND SUBMARINES: A COMPUTER PROGRAM

This report is part of a continuing study of the interaction of the underwater explosion shock wave with the ocean bottom. The computer program described in this paper was primarily written to calculate the translational velocity induced in surface ships by bottom reflected shock waves. These calculations provide a method of comparing the damage producing potential of the reflections for various bottom materials. The work was done under the supervision and cooperation of Dr. H. G. Snay (240).

This study was supported by the Defense Atomic Support Agency through Task DASA-MA 002-20 P. 106 (Energy Focussing and Refraction Effects).

GEORGE G. BALL
Captain, USN
Commander

C. J. ARONSON
By Direction

CONTENTS

ILLUSTRATIONS

THE PEAK TRANSLATIONAL VELOCITY OF SURFACE SHIPS AND SUBMARINES: A COMPUTER PROGRAM

1. INTRODUCTION

The peak translational velocity (PTV) of the center of gravity of a naval ship or submarine induced by underwater explosion shock waves is generally used to describe the degree of impairment of their mobility and weapon delivery capabilities. The model presently being used to calculate the PTV is that developed primarily for submarines by W. W. Murray (reference (1)). This model treats the interaction of an exponentially decaying acoustic plane wave with an infinitely long cylinder. For pulses of nuclear dimensions the assumption of plane incident waves is usually justified because the ranges considered are large compared to the dimensions of the ship or submarine.

In the application of Murray's theory to waves which have been reflected from the ocean bottom or refracted by velocity gradients in the ocean one encounters the need for calculating the PTV for wave shapes other than exponential. One of the best ways to make such a calculation for an arbitrary wave shape is through a superposition of step wave responses. Murray has calculated the step wave translational velocity curve and also the step wave acceleration. J. A. Goertner (in a confidential report) has written a computer program which uses Murray's curves to calculate the PTV for an arbitrary incident wave by decomposing the wave into a sum of step waves. This program has been used successfully in calculating the PTV of refracted waves, but is not well suited for bottom reflection studies.

In this paper Murray's theory is described briefly, and a computer program is explained which computes the PTV for an arbitrary wave shape in a somewhat different manner than Goertner's program. The incident pulse used in the program of this paper may have a singularity of the logarithmic type such as encountered in supercritical bottom reflections. The PTV is calculated by a convolution integral containing the incident wave shape and the step wave acceleration. The curve of the step wave acceleration has been recalculated so that the model can be more closely followed than is possible using Murray's curve. The theory is extended to surface ships, and the program calculates the PTV for both surfaced and submerged targets.

2. THEORY FOR CALCULATING THE TRANSLATIONAL VELOCITY OF AN INFINITELY LONG CYLINDER

2.1 Assumptions

Murray derived his equations for a rigid and neutrally buoyant cylinder of radius a. It is assumed that the displacement of the cylinder from its initial position is small compared to its radius. The equations were derived for athwartship attack; that is, the wave front is parallel to the longitudinal axis of the cylinder.

2.2 Translational Velocity of a Submerged Cylinder

Let the incident wave be given by

$$p(t) = p_F \exp \left[- (t - R/c_w)/G\right] , \; t > R/c_w \qquad (2.2.1)$$

where t is the time, R, the distance from the source to the target, G, the time constant of the exponential shock wave (usually denoted by θ), and c_w, the sound velocity of water. The peak pressure of the wave is p_F. For this exponential pulse Murray obtained the following equation for the translational velocity of a totally submerged cylinder

$$v(\tau) = - \frac{P_F}{\rho_w c_w} \frac{i}{\pi} \int_{-i\infty+\nu}^{i\infty+\nu} \frac{\exp\left[z(\tau-1)\right]}{z^2 \, (z+q) \, K_2 \, (z)} \, dz \; , \qquad (2.2.2)$$

where the integration variable z is a complex magnitude and ρ_w is the density of water. The symbol τ denotes the reduced time $\tau = c_w t/a$, and q is the reduced radius $q = a/c_w G$. For a step wave G becomes infinite, and we have q = 0. The function $K_2(z)$ is the modified Bessel Function of the second kind of the order two. The path of integration is to be taken in the right half of the complex plane, hence the constant ν must be real and positive. For practical purposes, a good choice of ν is unity.

2.3 Reduced Step Wave Acceleration

Upon differentiating $v(\tau)$ and setting q = 0, the desired expression for the reduced step wave acceleration of the cylinder is

$$A(\tau) = \left(\frac{\rho_w c_w}{p_F}\right) \frac{dv}{d\tau} = \frac{-i}{\pi} \int_{-i\infty+\nu}^{i\infty+\nu} \frac{\exp\left[z(\tau-1)\right]}{z^2 K_2(z)}\, dz \,. \qquad (2.3.1)$$

This function, calculated by the method described in Appendix A, is shown in Figure 2.3.1.

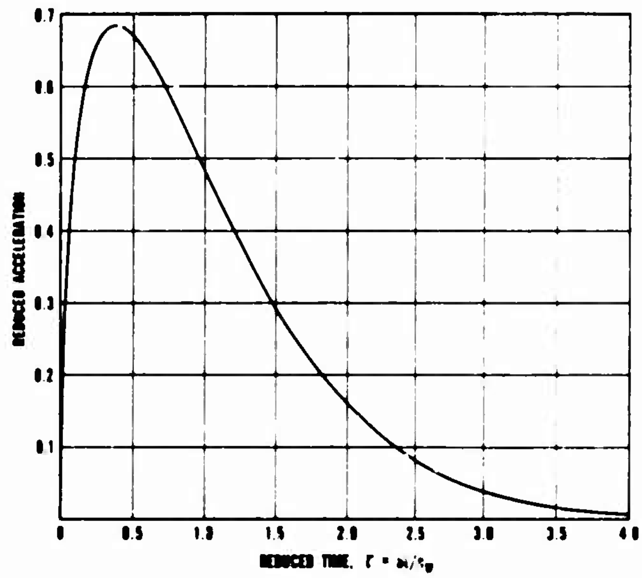

Figure 2.3.1 REDUCED STEP WAVE ACCELERATION OF A CYLINDER

2.4 Translational Velocity for an Arbitrary Wave Shape p(t)

The reduced step wave acceleration $A(\tau)$ plays the role of a Green's function for the problem. The translational velocity $V(\tau)$ from an arbitrary incident wave $p(t)$ can be written

$$V(\tau) = \frac{1}{\rho_w c_w} \int_0^{\tau} p(qa/c_w)\; A(\tau - q)\; dq \,, \qquad (2.4.1)$$

where $\tau = c_w t/a$. If the integration variable is changed so that it has the dimensions of time, $V(t)$ is then given by

$$V(t) = \frac{1}{\rho_w a} \int_0^{t} p(u)\, A(\tau - c_w u/a)\; du \,. \qquad (2.4.2)$$

This is the equation used to calculate V(t) in the PTV PROGRAM described in Section 3.

2.5 Translational Velocity of a Surface Ship

To apply the above equation to the response of a surface ship, two assumptions are made: (1) the target is considered to be a cylinder floating on the surface with its axis at the water line. (2) the vertical translational velocity is assumed to be twice the vertical component of the translational velocity the cylinder would acquire deeply submerged. The horizontal motion of the ship is not taken into account.

These assumptions are usually made for calculation of damage to surface ships, although it is realized that it may be an oversimplification. Effects such as cavitation are also ignored. This process is known to occur below ships and may be of importance.

Under the above assumptions, the vertical translational velocity of a floating cylinder when subjected to a pressure pulse p(t) is then

$$V_s(t) = 2V(t) \cos \alpha \, , \qquad (2.5.1)$$

and

$$V_s(t) = \frac{2 \cos \alpha}{\rho_w a} \int_0^t p(u) \, A(\tau - c_w u/a) \, du \, , \qquad (2.5.2)$$

where α is the angle between the plane wave front and a normal to the water surface, sometimes called the incident angle.

The program described in Section 3 calculates both V(t) and $V_s(t)$ and their maximum values, the peak translational velocities PTV.

2.6 Comparison of the Responses of a Target to Exponential and Supercritical Bottom Reflected Pulses.

The experimental data correlating the shock damage from an underwater explosion to the peak translational velocity, PTV, have been obtained for free water pulses or for free water pulses cut off by surface reflections. Both of these pulse shapes are initially exponential. Pulse shapes encountered in the study of supercritical bottom reflections are not exponentials, and the question arises whether the same shock damage results if the PTV's are the same. Two examples of these pulses, along with an exponential, are given in Figure 2.6.1. As shown in Figure 2.6.2 these pulses produce the same PTV on a cylinder of radius 22 ft.

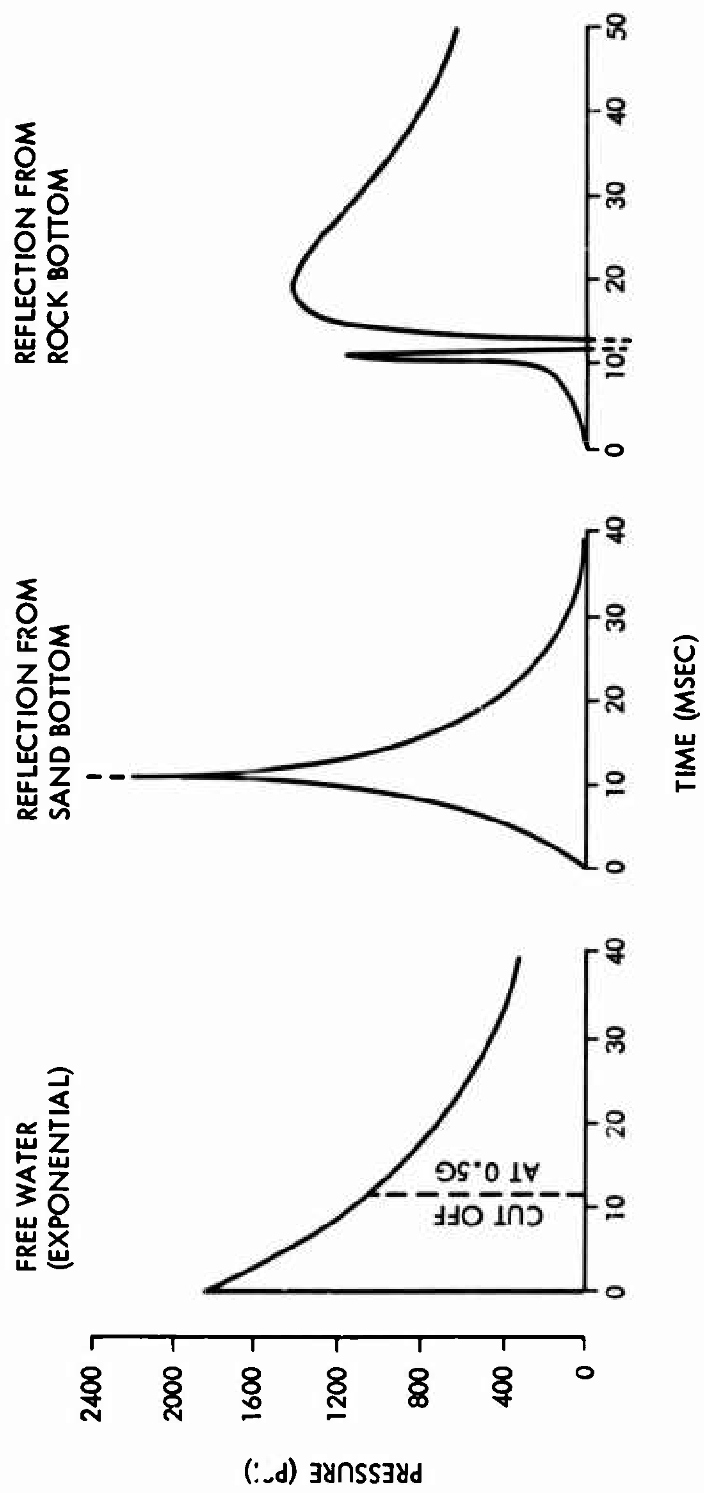

FIG. 2.6.1 FREE WATER AND BOTTOM REFLECTED PULSES PRODUCED BY A 10 KT CHARGE

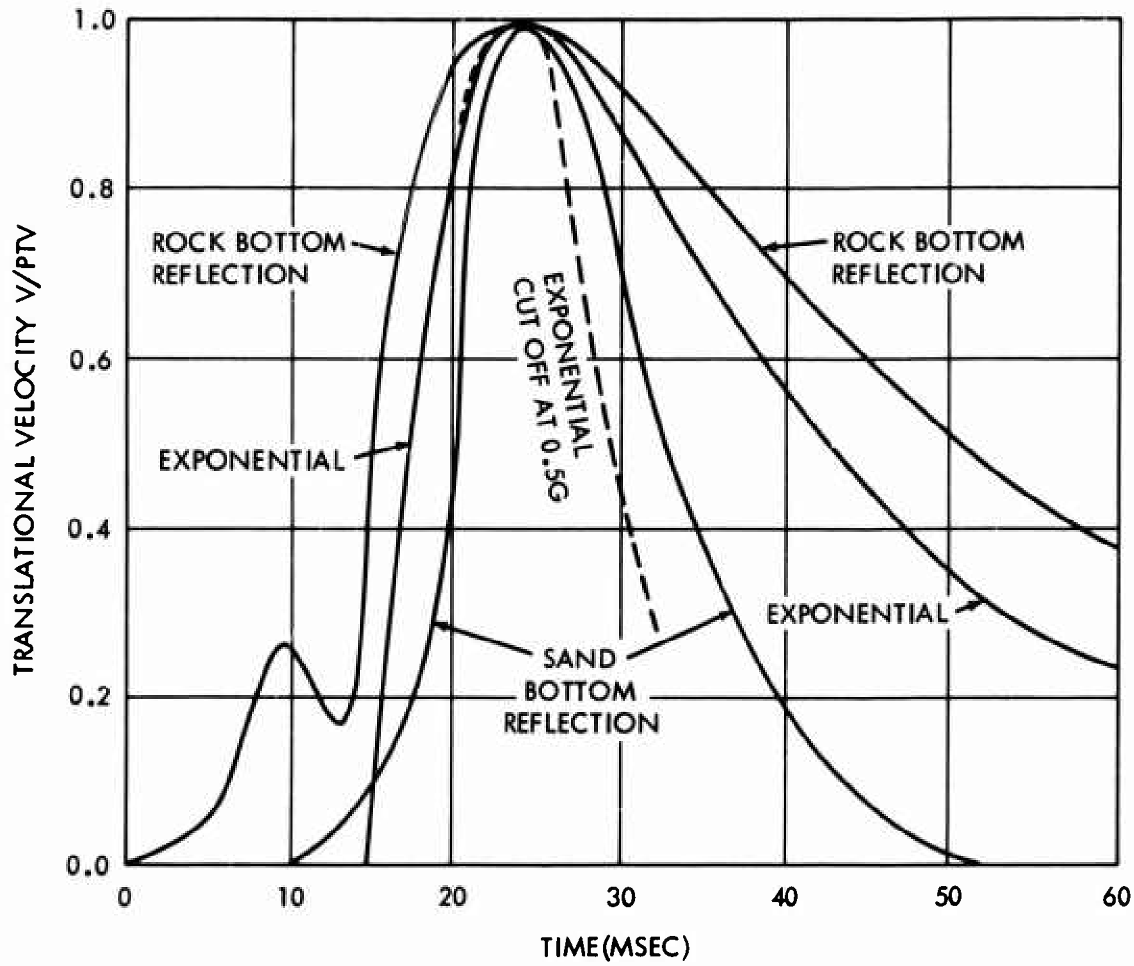

FIG. 2.6.2 RESPONSES OF A CYLINDER OF RADIUS 22 FT. TO FREE WATER AND BOTTOM REFLECTED PULSES PRODUCED BY A 10 KT CHARGE

Ignoring the early parts of the responses to the reflected pulses, these curves have roughly the same shape around the peak as the response to the exponential. After the peak the cut-off exponential response deviates much more than those of the bottom reflections. Having the same PTV and similar accelerations, the pulses of Figure 2.6.1 are expected to cause the same degree of damage. This means that PTV damage criteria derived for exponential pulses can also be applied to super-critical bottom reflections and other similar non-exponential pulses.

3. COMPUTER PROGRAM FOR CALCULATING PEAK TRANSLATIONAL VELOCITY

3.1 General Program Description

The peak translational velocity program or simply the PTV PROGRAM has been written in FORTRAN IV for the NOL CDC 6400 computer. A complete listing is given in Appendix C. This program calculates the PTV for both surface ships and submarines using the theory described in Section 2.

The PTV PROGRAM is composed of seven subroutines: PTV, FV, F1, XMAX, VTAB, PTAB, and FGI. The package is used by calling subroutine PTV from a main or executive program written by the user which supplies the pressure time history $p(t)$.

The PTV is obtained from equations (2.4.2) and 2.5.1). But in order that we may integrate numerically over a singularity in $p(t)$ at $t = t_c$, or $\tau = \tau_c = c_w t_c/a$, of the form $\ln|t-t_c|$, the integration variable u is changed as follows:

$$\text{for } u \le t_c \qquad w = (t_c - u)^{1/2}$$

$$\text{for } u \ge t_c \qquad z = (u - t_c)^{1/2} .$$

Equation (2.4.2) then becomes for $t > t_c$

$$V(\tau) = \frac{2}{\rho_w a} \left\{ - \int_{w(0)}^{w(t_c)} p(u)\, A(\tau - c_w u/a) w \, dw + \int_{z(t_c)}^{z(t)} p(u)\, A(\tau - c_w u/a) z \, dz \right\}. \qquad (3.1.1)$$

where $\tau = c_w t/a$. These integrals are evaluated in FUNCTION FV using the Gaussian quadrature of FUNCTION FGI. From V(t) we then obtain $V_s(t)$ using equation (2.5.1).

The $A(\tau)$ curve which has been calculated by the procedure of Appendix A is stored in the arrays QQX and QQY in FUNCTION F1. The reduced time τ is in QQX and A in QQY. The function $A(\tau - c_w u/a)$ is evaluated from these arrays by quadratic interpolation in FUNCTION VTAB. Similarly p(u) is determined by interpolation in VTAB of the arrays QX and QY which hold the time t in seconds and the incident pressure in psi. Near the singularity at $t = t_c$ the FUNCTION PTAB performs the quadratic interpolation for the pressure.

The convergence of the integrals in equation (3.1.1) is made possible because

$$\lim_{u \to t_c} w \ln|t - t_c| = \lim_{u \to t_c} z \ln|t-t_c| = 0. \qquad (3.1.2)$$

As implied in equation (3.1.1) the variables w and z are used for integration over the whole range of τ. Little difficulty is encountered in the numerical integration if the pressure pulse p(t) has no rapidly changing, high amplitude contributions far from the peak at $\tau_c = c_w t_c/a$.

The values of V(t) and $V_s(t)$ depend on the previous pressure history. Since $A(\tau)$ is very small for $\tau \gtrsim 8$, the integration range is restricted to at most from $u = \tau - 8$ to $u = \tau$. Thus if significant rapidly changing pulses occur away from τ_c by about $\tau = 8$, the PTV PROGRAM can be applied to each peak separately since the target response from one pulse is essentially damped out before the arrival of the next pulse. The actual PTV can then be found from the maximum of these results.

The maximum or peak values of V(t) and $V_s(t)$, the PTV's, are obtained as follows. An initial search for a maximum velocity is made from some $t = t_o$ to $t = t_1$. The values of t_o, t_1, and the number of steps are prescribed by the user in the call to subroutine PTV. Then several iterations are made around this maximum. Subroutine XMAX determines the maximum value of the translational velocity, but subroutine PTV controls the iteration and makes the calls to FUNCTION FV which sets up the integration for V(t). Iteration terminates when the relative difference between the two largest absolute values of V(t) is less than .001. If the iteration does not converge after five cycles, iteration is also terminated and a warning is printed. In either case values of the PTV for submerged targets, the maximum of V(t), and for floating targets, the maximum of $V_s(t)$, are returned to the main program.

3.2 <u>Use of the PTV PROGRAM</u>

To use the PTV PROGRAM subroutine package a main program must be set up by the user to supply the incident pulse p(t). The time in seconds and the pressure in psi must be stored in the arrays QX and QY as mentioned previously in Section 3.1. When the pressure history is short compared to the target transit time a/c_w, the PTV is likely to occur at a time beyond the last value of the pressure history. Thus to provide for extrapolation beyond the end of the actual pressure history the first unused storage of the QX array should be set to some very large value as 1.0E20. The corresponding QY storage should be set to zero or some other appropriate asymptotic value of p(t).

The QX and QY arrays are transferred to the PTV PROGRAM by COMMON storage. The statements COMMON /QXY/QX,QY and DIMENSION QX(1000), QY(1000) must be in the main program. In subroutines PTV and F1 the additional common storage is used: COMMON/QIS/IS. This statement is not needed in the main program.

Once the pressure history has been defined, the peak translational velocity is then obtained by calling subroutine PTV as follows:

CALL PTV (TIMER2, T3, T4, T5, RAD, PTS, OPTION, COSA, RHOW, CWAT, T, V, VS).

<u>INPUT</u> The following variables are inputs to subroutine PTV:

TIMER2 — Time t_c in seconds of the singularity or peak of the incident pulse. For a simple exponential pulse set TIMER2 = 0. The pressure at a singularity should be set to some number with absolute value greater than 1.0E20 as a signal to the interpolation subroutine PTAB.

T3 — Signals the approach of the singularity of the incident pulse p(t). If there is no singularity set T3 = TIMER2. When there is a singularity, T3 should have a value such that there are included at least two points of the QX array on each side of the singularity in the time interval $T3 < t < 2t_c - T3$.

T4 — Smallest time t_o at which the translational velocity is to be calculated. If the peak of p(t) occurs at or near zero, use T4 = 0. In other cases T4 (and T5 below) can be determined by remembering that the translational velocity at time t is calculated using the pressure history of the interval $t - 8a/c_w$ to t.

T5 — Largest time t_1 at which the translational velocity is to be calculated.

RAD — The cylinder radius a in feet.

PTS — The number of times at which the translational velocity is to be calculated in the initial search for the PTV. This search is made in the time interval $T4 \leq t \leq T5$. The maximum value PTS can be is 50.

OPTION — Controls printing in subroutine PTV. There is no printing if OPTION > 0. There is printing if OPTION ≤ 0.

COSA — cos α. See Section 2.5 for an explanation of α.

RHOW — Density of water ρ_w in gm/cm^3.

CWAT — Sound velocity of water c_w in ft/sec.

<u>OUTPUT</u> The following variables are outputs returned to the main program. When OPTIO[illegible], these results are printed out in subroutine PTV.

T — Time t in seconds. The time of the PTV is returned to the main program.

V — The translational velocity V(t) in ft/sec of a submerged target. The PTV is returned.

VS — The translational velocity $V_s(t)$ in ft/sec of a floating target. The PTV is returned.

A sample print out for a pressure pulse $p(t) = \exp(-125t)$, or $p(t) = \exp(-\tau)$ when $a/c_w = .008$, is shown in Appendix D. The input to subroutine PTV is included in the print out.

3.3 Important FORTRAN Symbols Not Included in the Call to Subroutine PTV

Dimensioned Variables

SUBROUTINE PTV

QX, QY — Time in seconds and pressure in psi of the incident pulse. These arrays must be defined in the user's executive program.

QQX, QQY — Reduced time and reduced acceleration of step wave. These arrays appear in FUNCTION F1.

IS(1)	Index for the beginning of the interpolation search in QQX array.
IS(2)	Index for the beginning of the interpolation search in QX array.
G	Array for transferring to FUNCTIONS FV and F1 variables in the integrand of V.
G(1) = T	Time t
G(2)	c_w/a
G(3)	t_c
G(4)	Signal for FUNCTION F1. In equation (3.1.1) G(4) = - 1.0 for the first integral and + 1.0 for the second integral.
G(5)	$(t_c - T3)^{1/2}$
G(6)	$w(0) = t_c^{1/2}$
A, C	Storage for time and V(t). Used by subroutine XMAX to determine the maximum $\lvert V(t) \rvert$ = C(M) and the next largest value C(M1).

<u>Non- ..ensioned Variables</u>

SUBROUTINE PTV

DT	Increm..nt of time.
M	See A and C above.
T1, T2, V1, V2	Temporary storages of A(M), A(M1), C(M), C(M1).
VS1	Value of $V_s(t)$ when V(t) = C(M1).

FUNCTION FV

N = 18	The integrations of equation (3.1.1) are performed using a four point Gaussian quadrature per subinterval of integration. N is the maximum number of subintervals allowed for the total integration interval.
NN, N1, NNN	The number of subintervals of integration used. NN is used if the total integration interval does not include t_c. N1 and NNN are used if t_c is included: N1 for the integration variable $u \leq \tau_c$ and NNN for $u > \tau_c$.
X	$t - 8c_w/a$ used to restrict integration to the interval $\tau - 8$ to τ.

Z1, Z2, Z3	Limits of integration in equation (3.1.1). In the calls to FUNCTION FGI the first variable is the lower limit of integration, the second is the upper limit.
FV	The sum of the integrals of equation (3.1.1).
FUNCTION F1	
Z	Integration variables w and z.
X	Time corresponding to integration variable u.
XD	Reduced time equal to $c_w \cdot (t - u)/a$.
P	Interpolated pressure at time X.
F1	Integrands of equation (3.1.1).

REFERENCES

1. Murray, W. W., "Interaction of a Spherical Acoustic Wave with a Beam of Circular Cross Section," Underwater Explosions Research Division, Norfolk Naval Shipyard, UERD Report 1-55, Jan 1955, Unclassified.

2. Abramowitz, M., and Stegun, I. A., "Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables," Dover Publications, New York, 1965

3. Jahnke, E., Emde F., and Lösch, F., "Tables of Higher Functions," McGraw-Hill, New York, 1960.

APPENDIX A

CALCULATION OF THE REDUCED STEP WAVE ACCELERATION A(τ)

In order to evaluate A(τ) from equation (2.3.1) it is necessary to transform the integral in the complex plane to a real integral. Murray has accomplished this transformation by using a series expansion when τ is small, up to about $\tau = 1$, and contour integration at larger values of τ. However, to obtain a more accurate A(τ), we have used the more direct approach explained below.

The integration variable z can be written $z = x + iy$ for x and y real. If the integration path is taken along the line $x = \gamma = 1$, z becomes $z = 1 + iy$. The complex functions in the integrand of A(τ) can then be separated into their real and imaginary parts:

$$z^2 = (1 - y^2) + i2y ,$$

$$\exp\left[z(\tau-1)\right] = \exp(\tau-1)\cos\left[y(\tau-1)\right] + i\exp(\tau-1)\sin\left[y(\tau-1)\right] ,$$

and
$$K_2(z) = \mathrm{Re}(K_2) + i\,\mathrm{Im}(K_2) ,$$

where $\mathrm{Re}(K_2)$ and $\mathrm{Im}(K_2)$ denote the real part and the imaginary part of $K_2(z)$. Explicit expressions from which $\mathrm{Re}(K_2)$ and $\mathrm{Im}(K_2)$ can be obtained will be given later. On substituting the above functions in A(τ), equation (2.3.1), and then separating real and imaginary parts of the integrals one obtains

$$A(\tau) = \frac{\exp(\tau-1)}{\pi}\left\{\int_{-\infty}^{\infty}\frac{E_1\cos\left[y(\tau-1)\right] + E_2\sin\left[y(\tau-1)\right]}{E_1^{\,2} + E_2^{\,2}}\,dy\right.$$

$$\left. + i\int_{-\infty}^{\infty}\frac{E_1\sin\left[y(\tau-1)\right] - E_2\cos\left[y(\tau-1)\right]}{E_1^{\,2} + E_2^{\,2}}\,dy\right. , \qquad (A.1)$$

where

$$E_1 = (1 - y^2)\,\mathrm{Re}(K_2) - 2y\,\mathrm{Im}(K_2) \qquad (A.2)$$

and

$$E_2 = (1 - y^2)\,\mathrm{Im}(K_2) + 2y\,\mathrm{Re}(K_2) \ . \qquad (A.3)$$

A substitution of -y for y in equation (A.1) shows that the integrand of the first integral of A(τ) is even and the integrand of the second integral is odd.

Hence the second integral is zero and $A(\tau)$ is a real function which can be written

$$A(\tau) = \frac{2}{\pi} \exp(\tau - 1) \int_0^\infty \frac{E_1 \cos[y(\tau-1)] + E_2 \sin[y(\tau-1)]}{E_1^2 + E_2^2} dy . \qquad (A.4)$$

For $y < 15$ we have calculated $K_2(z)$ from the expression

$$K_2(z) = \int_0^\infty \exp(-z \cosh \Phi) \cosh 2\Phi \, d\Phi . \qquad (A.5)$$

Separating the exponential into its real and imaginary parts and substituting $z = 1 + iy$, we obtain

$$K_2(z) = \int_0^\infty \exp(-\cosh \Phi) \cos(y \cosh \Phi) \cosh 2\Phi \, d\Phi$$

$$- i \int_0^\infty \exp(-\cosh \Phi) \sin(y \cosh \Phi) \cosh 2\Phi \, d\Phi . \qquad (A.6)$$

Substitution of this expression for $K_2(z)$ into equations (A.2) and (A.3) yields the following expressions for E_1 and E_2

$$E_1 = \int_0^\infty \left[(1-y^2) U + 2y Z\right] U_1 \, d\Phi \qquad (A.7)$$

$$E_2 = \int_0^\infty \left[2y U - (1-y^2) Z\right] U_1 \, d\Phi \qquad (A.8)$$

where: $U = \cos(y \cosh \Phi)$

$Z = \sin(y \cosh \Phi)$

$U_1 = \exp(-\cosh \Phi) \cosh 2\Phi$.

These integrals converge very rapidly because of the expression U_1 which approaches zero like $\exp(-\exp \Phi)$ for about $\Phi = 4$ or larger. It is shown in Appendix B that the error in truncating the integration in E_1 and E_2 at $\Phi = 4.5$ is less than 1 part in 10^{13}.

Even though the integrals converge rapidly, they become increasing more difficult to evaluate numerically as y increases because of the oscillatory factors U and Z. At about y = 15 an asymptotic expansion for evaluating $K_2(z)$ becomes more practical.

For y between 15 and 1000, the following asymptotic expansion (reference (2)) of $K_2(z)$ is used

$$K_2(z) \approx \left(\frac{\pi}{2z}\right)^{1/2} \exp(-z) \left[1 + \frac{16-1^2}{1!\ 8z} + \frac{(16-1^2)(16-3^2)}{2!\ (8z)^2} + \ldots\right]. \quad (A.9)$$

where again z = 1 + iy. Near y = 15 nine terms of the series in brackets are used, i.e., the lowest ordered term used is of the order $1/z^8$. Retaining nine terms insures that the series truncation error for y = 15 is less than 2×10^{-9}. Between y = 15 and y = 1000 fewer terms are needed for larger y; however, a sufficient number of terms are retained so that the truncation error is less than that at y = 15.

The integral for $A(\tau)$ from y = 1000 to infinity is calculated from an approximate equation obtained by neglecting terms of order $1/y^2$ or smaller compared to one. From equation (A.9) the approximate relation for $K^2(1 + iy)$ is obtained

$$K_2(1+iy) \approx \left[\frac{\pi}{2y}\right]^{1/2} \exp(-1) \left[\cos(y+\psi) - i \sin(y+\psi)\right] \left[1 - \frac{115}{8y}\right], \quad (A.10)$$

where

$$\cos\psi \approx \frac{1}{\sqrt{2}}\left(1 + \frac{1}{2y}\right) \text{ and } \sin\psi \approx \frac{1}{\sqrt{2}}\left(1 - \frac{1}{2y}\right). \quad (A.11)$$

Substituting the real and imaginary parts of K_2 from equation (A.10) into equations (A.2) and (A.3) and neglecting terms of order $1/y^2$, the following relations are obtained:

$$E_1 \approx \left[\frac{\pi}{2y}\right]^{1/2} \exp(-1) \left[-y^2 \cos(y + \psi) + \frac{31}{8} y \sin(y + \psi)\right], \quad (A.12)$$

$$E_2 \approx \left[\frac{\pi}{2y}\right]^{1/2} \exp(-1) \left[y^2 \sin(y + \psi) + \frac{31}{8} y \cos(y + \psi)\right], \quad (A.13)$$

and

$$E_1^2 + E_2^2 \approx \frac{\pi}{2} y^3 \exp(-2). \quad (A.14)$$

Combining the above equations with equation (A.4), the remainder R(a) of the A(τ) integral from y = a to infinity is

$$R(a) \approx \left[\frac{2}{\pi}\right]^{3/2} \exp(\tau) \int_a^\infty y^{-3/2} \left[\frac{31}{8y} \sin(\tau y + \psi) - \cos(\tau y + \psi)\right] dy \quad , \qquad (A.15)$$

where the trignometric relations for the sine and cosine of the sum of two angles have been used and where the lower ordered terms have been neglected. For the numerical computations a = 1000 is used.

Using similar manipulations as above and substituting cos ψ and sin ψ from equation (A.11), R(a) can be written

$$R(a) \approx \frac{2}{\pi^{3/2}} \exp(\tau) \int_a^\infty y^{-3/2} \left[\left(1 + \frac{27}{8y}\right) \sin \tau y - \left(1 - \frac{27}{8y}\right) \cos \tau y\right] dy. \qquad (A.16)$$

Integration by parts can then be used to obtain

$$R(a) \approx \frac{\exp(\tau)}{\pi^{3/2}} \Big\{(4 - 9\tau)\left[a^{-1/2} (\sin a\tau - \cos a\tau) + \sqrt{2\pi\tau}\,(1 - S(a\tau)\right.$$

$$\left. - C(a\tau))\right] + \frac{9}{2} a^{-3/2} \left[\sin a\tau + \cos a\tau\right]\Big\} , \qquad (A.17)$$

where S(aτ) and C(aτ) are commonly called Fresnel's integrals and are defined

$$S(a\tau) = \frac{1}{\sqrt{2\pi}} \int_0^{a\tau} \frac{\sin x}{\sqrt{x}} dx \text{ and } C(a\tau) = \frac{1}{\sqrt{2\pi}} \int_0^{a\tau} \frac{\cos x}{\sqrt{x}} dx \quad . \qquad (A.18)$$

These integrals have the following asymptotic expansions (see reference (3)):

$$C(z) \approx \frac{1}{2} + \frac{\sin z}{\sqrt{2\pi z}} \left[1 - \frac{1 \cdot 3}{(2z)^2} + \frac{1 \cdot 3 \cdot 5 \cdot 7}{(2z)^4} - \ldots\right]$$

$$- \frac{\cos z}{\sqrt{2\pi z}} \left[\frac{1}{2z} - \frac{1 \cdot 3 \cdot 5}{(2z)^3} + \ldots\right] \qquad (A.19)$$

$$S(z) \approx \frac{1}{2} - \frac{\cos z}{\sqrt{2\pi z}} \left[1 - \frac{1 \cdot 3}{(2z)^2} + \frac{1 \cdot 3 \cdot 5 \cdot 7}{(2z)^4} - \ldots\right]$$

$$- \frac{\sin z}{\sqrt{2\pi z}} \left[\frac{1}{2z} - \frac{1\cdot3\cdot5}{(2z)^3} + \ldots \right] \quad . \tag{A.20}$$

Substitution of the above equations into equation (A.17) gives

$$R(a) \sim \frac{\exp(\tau)}{a^{1/2}\pi^{3/2}} \left\{ (4 - 9\tau) \quad (\sin a\tau - \cos a\tau)\left(\frac{1\cdot3}{(2a\tau)^2} - \frac{1\cdot3\cdot5\cdot7}{(2a\tau)^4} + \ldots\right) \right.$$

$$\left. + (\sin a\tau + \cos a\tau) \left[\left(\frac{1}{2a\tau} - \frac{1\cdot3\cdot5}{(2a\tau)^3} + \ldots\right) + \frac{9}{2a} \right] \right\} . \tag{A.21}$$

In summary, to evaluate $A(\tau)$ from equation (A.4) E_1 and E_2, which are defined by equations (A.2) and (A.3), are given by equations (A.7) and (A.8) for $0 < y \le 15$ and obtained from equation (A.9) for $15 \le y \le 1000$. The integral from $y = 1000$ to infinity is $R(a)$, equation (A.21), where $a = 1000$. The $A(\tau)$ curve shown in figure 2.3.1 was calculated by the above method on the NOL IBM 7090 computer. A table giving the $A(\tau)$ array to six decimal places is contained in the DATA statement of FUNCTION F1 of the FORTRAN listing of the PTV PROGRAM which is given in Appendix C.

APPENDIX B

CONVERGENCE OF THE INTEGRALS E_1 AND E_2

It is the object of this section to show that the improper integrals, equations (A.7) and (A.8), used to obtain E_1 and E_2 are convergent. We also obtain an upper bound on the error introduced by stopping the integration to infinity at a finite value of the integration variable Φ.

It is commonly proved in text books of integral calculus that improper integrals of the form of E_1 and E_2 are convergent if the integral of the absolute value of the integrand is convergent. The converse does not necessarily hold. Denote the integrand of E_1 by E_1' and that of E_2 by E_2'. Since in general $|a \pm b| \leq |a| + |b|$, $|\sin a| \leq 1$, $|\cos a| \leq 1$, and $|ab| = |a|\,|b|$; we obtain

$$\left|E_1'\right| = \left|[(1 - y^2)\, U + 2y\, Z]\;\; U_1\right|$$

$$\leq (1 + y^2 + 2y) \exp\,(-\cosh \Phi) \cosh 2\Phi. \tag{B.1}$$

This result also holds for E_2'. Since

$$\cosh 2\Phi = [\exp\,(2\Phi) + \exp\,(-2\Phi)]/2 < \exp\,(2\Phi),$$

the above inequality can be simplified to

$$\left|E_1'\right| < (1 + y^2 + 2y) \exp\,(2\Phi - \cosh \Phi)\;. \tag{B.2}$$

At $\Phi = 4.5$, $\cosh \Phi \sim 45.01$. Hence for $\Phi \geq 4.5$ we find $2\Phi - \cosh \Phi < -36 = -8\Phi$. Expression (B.2) becomes for $\Phi \geq 4.5$

$$E_1' < (1 + y^2 + 2y) \exp\,(-8\Phi)\;. \tag{B.3}$$

Integrating this expression leads to

$$\int_{4.5}^{\infty} E_1'\, d\Phi < \int_{4.5}^{\infty} E_1'\;\; d\Phi < \frac{1}{8}\,(1 + y^2 + 2y)\, e^{-36}\;\;. \tag{B.4}$$

For the range $y \leq 15$ in which E_1 is calculated from equation (A.7), we can be assured that

$$\int_{4.5}^{\infty} E_1' \, d\Phi < 1 \times 10^{-13} \,, \qquad \text{(B.5)}$$

which shows integrating to $\Phi = 4.5$ is quite sufficient because the value of $\left| E_1 \right|$ in this range of y is about 1 to 10. Since E_1' has no singularities in $0 \le \Phi < 4.5$ and since the integral from $\Phi = 4.5$ to infinity is finite (and very small) we can conclude that E_1 is convergent.

All of the steps after expression (B.1) hold for E_2 as well as E_1. Consequently, the inequality (B.5) holds also for E_2 and convergence follows.

APPENDIX C

FORTRAN IV LISTING OF PTV PROGRAM

```
C      *****  PTV PROGRAM  *****
C
       SUBROUTINE PTV(TIMER2,T3,T4,T5,RAD,PTS,OPTION,COSA,RHOW,CWAT,
      1 T,V,VS)
C
C      THIS SUBPROGRAM CONTROLS THE ITERATION FOR THE PEAK
C      TRANSLATIONAL VELOCITY, PTV.  IT IS THE ONLY SUBROUTINE OF THE
C      PTV PROGRAM WHICH IS CALLED FROM THE MAIN PROGRAM.
C
       DIMENSION QX(1000),QY(1000),IS(2)
       DIMENSION G(6)
       DIMENSION A(50),C(50)
       COMMON /QXY/QX,QY
       COMMON /QIS/IS
C
C
       IF(OPTION.GT.0.) GO TO 10
       WRITE(6,580)
       WRITE(6,600) TIMER2,T3,T4,T5,RAD,PTS,OPTION,COSA,RHOW,CWAT
       WRITE(6,590)
C
C      74.21457 IS A UNITS CONVERSION FACTOR
    10 VC=2.*74.21457/RHOW/RAD
       N=PTS
       T=T4
       DT=(T5-T)/FLOAT(N-1)
       IF(T.LE.0.) N=N-1
       IF(T.LE.0.) T=DT/2.
       IS(1)=2
       IS(2)=1
       G(2)=CWAT/RAD
       G(3)=TIMER2
       G(5)=SQRT(TIMER2-T3)
       G(6)=SQRT(TIMER2)
C      INITIAL SEARCH FOR MAXIMUM VELOCITY
       DO 40 I=1,N
       G(1)=T
       V=VC*FV(G)
       A(I)=T
       C(I)=V
       VS=2.*COSA*V
       IF(OPTION.LE.0.) WRITE(6,610) T,V,VS
       T=T+DT
```

```
   40 CONTINUE
C     ITERATION FOR PTV
C     DETERMINE THE MAXIMUM VELOCITY FROM C ARRAY
      CALL XMAX(C,N,M,M1)
      A2=A(M1)
      C2=C(M1)
      A(1)=A(M)
      C(1)=C(M)
      A(2)=A2
      C(2)=C2
      DA=DT
      T=A(1)-1.8*DA
      IF(T.LE.0.) T=DA/5.
      DT=DA/2.
      DO 45 I=3,10
      G(1)=T
      V=VC*FV(G)
      A(I)=T
      C(I)=V
      VS=2.*COSA*V
      IF(OPTION.LE.0.) WRITE(6,610) T,V,VS
      T=T+DT
   45 CONTINUE
      N=10
      IF(IABS(M-M1).LT.3) GO TO 55
      T=A(2)-0.8*DA
      IF(T.LE.0.) T=DA/5.
      DT=DA/3.
      DO 50 I=11,16
      G(1)=T
      V=VC*FV(G)
      A(I)=T
      C(I)=V
      VS=2.*COSA*V
      IF(OPTION.LE.0.) WRITE(6,610) T,V,VS
      T=T+DT
   50 CONTINUE
      N=16
   55 CONTINUE
      DO 75 JJ=1,6
      CALL XMAX(C,N,M,M1)
      IF(JJ.LT.3) GO TO 62
      IF(ABS((C(M)-C(M1))/C(M)).LT.0.001) GO TO 110
      IF(JJ.EQ.6) GO TO 120
   62 N=10
      T1=A(M)
      T2=A(M1)
      V1=C(M)
      V2=C(M1)
      A(9)=T1
      A(10)=T2
      C(9)=V1
      C(10)=V2
```

```
      DT=ABS(T1-T2)/5.
      II=1
      DO 70 I=1,M
      T=T1+DT*FLOAT((I-10)/2*II)
      IF (T .LE. 0.0) GO TO 64
      G(1)=T
      V=VC*FV(G)
      VS=2.*COSA*V
      GO TO 66
C     WHEN T IS LESS THAN ZERO SET TO ZERO.
   64 T = 0.0
      V = 0.0
   66 IF (OPTION .LE. 0.0) WRITE (6,610) T,V,VS
      A(I)=T
      C(I)=V
      II=-1*II
   70 CONTINUE
   75 CONTINUE
  110 V=C(M)
      T=A(M)
      VS=2.*COSA*C(M)
      IF(OPTION.LE.0.) WRITE(6,620) A(M),C(M),VS
      RETURN
  120 V=C(M)
      T=A(M)
      VS=2.*COSA*C(M)
      VS1=2.*COSA*C(M1)
      WRITE(6,630) T,V,VS,A(M1),C(M1),VS1
      RETURN
C
C
  580 FORMAT(1H1,10X,30HTRANSLATIONAL VELOCITY PROGRAM )
  590 FORMAT(1H0,5X,45HITERATION FOR PEAK TRANSLATIONAL VELOCITY PTV //
     1 12X,9HTIME(SEC),8X,16HVELOCITY(FT/SEC) ,3X,25HVERTICAL VELOCITY(F
     2T/SEC) /29X,16HTARGET SUBMERGED,7X,17HTARGET AT SURFACE )
  600 FORMAT(1H0,5X,23HINPUT TO SUBROUTINE PTV // 10X,
     1 45HTIMER2,T3,T4,T5,RAD,PTS,OPTION,COSA,RHOW,CWAT //1P5E14.5/
     2 1P5E14.5 )
  610 FORMAT(1P3E22.6)
  620 FORMAT(1H0,6X,20H******************** ,9X,3HPTV,19X,3HPTV//
     1 1P3E22.6)
  630 FORMAT(1H0,42H*** WARNING ITERATION DID NOT CONVERGE *** ,5X,
     1 35HMAXIMUM AND NEAREST VALUE ARE GIVEN //
     1 12X,9HTIME(SEC),8X,16HVELOCITY(FT/SEC) ,3X,25HVERTICAL VELOCITY(F
     2T/SEC) /29X,16HTARGET SUBMERGED,7X,17HTARGET AT SURFACE /
     3 (1P3E22.6))
C
      END
```

```
      FUNCTION FV(G)
C
C     THIS SUBPROGRAM SETS UP THE INTEGRATION FOR
C     THE TRANSLATIONAL VELOCITY V
C
      DIMENSION G(6)
      EXTERNAL F1
      DATA N/18/
C
      NN=FLOAT(N)*G(1)*G(2)/8.
      NN=MAX0(NN,8)
      NN=MIN0(NN,N)
      X=G(1)-8./G(2)
      IF(X.GT.G(3)) GO TO 43
      Z1=G(6)
      IF(X.GT.0.) Z1=SQRT(G(3)-X)
      IF(G(1).GT.G(3)) GO TO 40
      G(4)=-1.0
      Z2=SQRT(G(3)-G(1))
C     INTEGRATION FOR T .LE. TIMER2
      FV=-FGI(Z1,Z2,NN,F1,G)
      RETURN
   40 Z2=0.
      Z3=SQRT(G(1)-G(3))
      IF(G(3).EQ.0.) GO TO 45
      G(4)=-1.0
      N1=Z1/(Z1+Z3)*FLOAT(NN)+2.0
      NNN=Z3/(Z1+Z3)*FLOAT(NN)+2.0
C     INTEGRATION FOR INTERVAL WHICH INCLUDES TIMER2
      V1=-FGI(Z1,Z2,N1,F1,G)
      G(4)=1.0
      V2=FGI(Z2,Z3,NNN,F1,G)
      FV=V1+V2
      RETURN
   43 Z2=SQRT(X-G(3))
      Z3=SQRT(G(1)-G(3))
   45 G(4)=1.0
C     INTEGRATION FOR T LARGER THAN TIMER2 BUT THE
C     INTERVAL DOES NOT INCLUDE TIMER2.
      FV=FGI(Z2,Z3,NN,F1,G)
      RETURN
      END
```

```
      FUNCTION F1(Z,G)
C
C     THIS SUBPROGRAM CALCULATES THE PRODUCT INCIDENT PRESSURE *
C     REDUCED STEP WAVE ACCELERATION BY CALLING THE INTERPOLATION
C     PROGRAMS VTAB AND PTAB.
C
      DIMENSION QX(1000),QY(1000),IS(2)
      DIMENSION G(6),QQX(120),QQY(120)
      COMMON /QXY/QX,QY
      COMMON /QIS/IS
C
C     REDUCED STEP WAVE ACCELERATION OF A CYLINDER
C
      DATA (QQX(I),I=1,106) /0.,.0125,.025,.0375,.050,.075,.100,
     1 .125,.150,.175,.200,.225,.250,.275,.300,.325,.350,.375,
     2 .4000,.425,.450,.475,.500,.525,.550,.575,.600,.625,.650,
     3 .675,.700,.725,.750,.775,.800,.825,.850,.875,.900,.925,.950,
     4 .975,1.00,1.05,1.10,1.15,1.20,1.25,1.30,1.35,1.40,1.45,
     5 1.50,1.55,1.60,1.65,1.70,1.75,1.80,1.85,1.90,1.95,2.00,
     6 2.05,2.10,2.15,2.20,2.25,2.30,2.35,2.40,2.45,2.50,2.55,
     7 2.60,2.65,2.70,2.75,2.80,2.85,2.90,3.00,3.10,3.2,3.3,3.4,
     8 3.5,3.6,3.7,3.8,3.9,4.0,4.2,4.4,4.6,4.8,5.0,5.25,5.50,
     9 5.75,6.00,6.25,6.5,7.0,7.5,8.0 /
      DATA (QQY(I),I=1,60) / 0.0, .198193,.275935,.332694,.378180,
     1 .448836,.502189,.544000,.577342,.604111,.625589,.642701,
     2 .656143,.666457,.674079,.679365,.682612,.684070,.683955,
     3 .682452,.679721,.675904,.671127,.665499,.659120,.652078,
     4 .644453,.636315,.627730,.618755,.609444,.599844,.589999,
     5 .579949,.569730,.559374,.548913,.538372,.527777,.517151,
     6 .506515,.495887,.485284,.464715,.443417,.422977,.402968,
     7 .383447,.364460,.346042,.328218,.311008,.294424,.278471,
     8 .263152,.248465,.234404,.220960,.208124,.195881 /
      DATA (QQY(I),I=61,106) / .184219,.173122,.162573,.152555,
     1 .143051,.134041,.125509,.117435,.109801,.102590,.095782,
     2 .089361,.083308,.077608,.072242,.067196,.062453,.057999,
     3 .053818,.049897,.046221,.039556,.033725,.028637,.024209,
     4 .020368,.017044,.014177,.011712,.009599,.007795,.006260,
     5 .003863,.002172,.001009,.000230,-0.000267,-0.000619,
     6 -0.000774,-0.000804,-0.000767,-0.000696,-0.000606,
     7 -0.000430,-0.000297,-0.000206 /
C
      IF(G(4).GT.0.) GO TO 20
      X=G(3)-Z*Z
      GO TO 30
   20 X=G(3)+Z*Z
   30 XD=(G(1)-X)*G(2)
      IF(Z.GT.G(5)) GO TO 35
      P=PTAB(X,QX,QY,IS(2))
      GO TO 40
   35 P=VTAB(X,QX,QY,IS(2))
   40 F1=Z*P*VTAB(XD,QQX,QQY,IS(1))
      RETURN
      END
```

```
      SUBROUTINE XMAX(B,N,M,M1)
C
C
C     THIS SUBPROGRAM DETERMINES THE LOCATIONS OF THE TWO LARGEST
C     ABSOLUTE VALUES OF MEMBERS OF THE B ARRAY.
C
C
      DIMENSION B(50)
      X=ABS(B(1))
      M=1
      DO 10 I=2,N
      IF(ABS(B(I)).LT.X) GO TO 10
      M=I
      X=ABS(B(M))
   10 CONTINUE
      M1=1
      IF(M.EQ.1) M1=2
      X=ABS(B(M1))
      DO 20 I=2,N
      IF(ABS(B(I)).LT.X) GO TO 20
      IF(I.EQ.M) GO TO 20
      M1=I
      X=ABS(B(M1))
   20 CONTINUE
      RETURN
      END
```

```
      FUNCTION VTAB(X,Y,Z,K)
C
C     THIS SUBPROGRAM PERFORMS A SECOND ORDER LAGRANGIAN INTERPOLATION
C
C     THE INDEPENDENT VARIABLF IS STORED IN THE Y ARRAY IN INCREASING
C     ORDER.   THE DEPENDENT VARIABLE IS STORED IN THE Z ARRAY.
C     X IS THE POINT AT WHICH THE FUNCTION IS TO BE EVALUATED.
C     K IS THE NUMBER OF THE ELEMENT IN THE Y ARRAY WHICH IS FIRST
C     COMPARED WITH X.
C
C
      DIMENSION Y(1000),Z(1000)
      IF(X.LE.0.) GO TO 50
      DO 10 I=K,1000
      J=I
      IF(Y(I).GT.X) GO TO 20
   10 CONTINUE
   20 J=MAX0(3,J-1)
      DO 30 I=1,1000
      IF(Y(J).LT.X) GO TO 40
      J=J-1
      IF(J.LT.3) GO TO 40
   30 CONTINUE
   40 K=J+1
      IF(Z(J).EQ.Z(K)) GO TO 60
      L=J-1
      A=(X-Y(K))/(Y(J)-Y(L))
      C=(X-Y(L))/(Y(K)-Y(J))
      IF((A.LT.-5.0).OR.(C.GT.5.0)) GO TO 60
      B=(X-Y(J))/(Y(K)-Y(L))
      VTAB=C*(B*Z(K)-A*Z(J))+A*B*Z(L)
      RETURN
   50 VTAB=0.
      RETURN
   60 VTAB=Z(J)+(X-Y(J))*(Z(K)-Z(J))/(Y(K)-Y(J))
      RETURN
      END
```

```
      FUNCTION PTAB(X,Y,Z,K)
C
C
C     THIS SUBPROGRAM PERFORMS A SECOND ORDER LAGRANGIAN INTERPOLATION
C     WITH PROVISIONS FOR HANDLING A SINGULARITY.
C     FUNCTION ARGUMENTS ARE THE SAME AS IN VTAB .
C
C
      DIMENSION Y(1000),Z(1000)
      IF(X.LE.0.) GO TO 50
      DO 10 I=K,1000
      J=I
      IF(Y(I).GT.X) GO TO 20
   10 CONTINUE
   20 J=MAX0(3,J-1)
      DO 30 I=1,1000
      IF(Y(J).LT.X) GO TO 40
      J=J-1
      IF(J.LT.3) GO TO 40
   30 CONTINUE
   40 J=J+1
      JJ=J
C
C     THE FOLLOWING THREE STATEMENTS PROVIDE FOR EXTRAPOLATION
C     AROUND A SINGULARITY.
C
      IF(ABS(Z(J)).GT.1.0E20)JJ=J-2
      IF(ABS(Z(J-1)).GT.1.0E20)JJ=J+1
      IF((JJ.EQ.J).AND.(ABS(Z(J-2)).LT.1.0E20)) JJ=J-1
C
      J=JJ
      K=J+1
      IF(Z(J).EQ.Z(K)) GO TO 60
      L=J-1
      A=(X-Y(K))/(Y(J)-Y(L))
      C=(X-Y(L))/(Y(K)-Y(J))
      IF((A.LT.-5.0).OR.(C.GT.5.0)) GO TO 60
      B=(X-Y(J))/(Y(K)-Y(L))
      PTAB=C*(B*Z(K)-A*Z(J))+A*B*Z(L)
      RETURN
   50 PTAB=0.
      RETURN
   60 PTAB=Z(J)+(X-Y(J))*(Z(K)-Z(J))/(Y(K)-Y(J))
      RETURN
      END
```

```
      FUNCTION  FGI(A,B,K,F,P)
C
C
C     THIS SUBPROGRAM INTEGRATES THE FUNCTION F BETWEEN THE LIMITS
C     A AND B USING A FOUR-POINT GAUSSIAN QUADRATURE IN EACH OF THE
C     K SUBINTERVALS.
C
C
      DIMENSION V(4),W(2),SUM(4),P(1)
      DATA V/ -.861136311594053,-.339981043584856,
     1 .339981043584856,.861136311594053 /
      DATA W/ .347854845137454,.652145154862546 /
      SUM(1)=0.0
      SUM(2)=0.0
      SUM(3)=0.0
      SUM(4)=0.0
      H=(B-A)/FLOAT(K)
      H2=H/2.
      AA=A+H2
      DO 20 L=1,K
      DO 10 I=1,4
      X=H2*V(I)+AA
   10 SUM(I)=SUM(I)+F(X,P)
   20 AA=AA+H
      FGI=H2*(W(1)*(SUM(1)+SUM(4))+W(2)*(SUM(2)+SUM(3)))
      RETURN
      END
```

APPENDIX D

SAMPLE PROGRAM OUTPUT FOR $p(t) = \exp(-125t)$

TRANSLATIONAL VELOCITY PROGRAM

INPUT TO SUBROUTINE PTV

TIMER2,T3,T4,T5,RAD,PTS,OPTION,COSA,RHOW,CWAT

0.	0.	0.	7.84000E-02	2.20000E+01
5.00000E+00	0.	8.66030E-01	1.00000E+00	5.00000E+03

ITERATION FOR PEAK TRANSLATIONAL VELOCITY PTV

TIME(SEC)	VELOCITY(FT/SEC) TARGET SUBMERGED	VERTICAL VELOCITY(FT/SEC) TARGET AT SURFACE
9.800000E-03	6.787638E-03	1.175660E-02
2.940000E-02	7.162470E-04	1.240583E-03
4.900000E-02	6.096742E-05	1.055992E-04
6.860000E-02	5.260610E-06	9.111692E-06
3.920000E-03	6.288036E-03	1.089126E-02
1.372000E-02	4.843787E-03	8.389729E-03
2.352000E-02	1.514464E-03	2.623143E-03
3.332000E-02	4.347603E-04	7.530309E-04
4.312000E-02	1.271495E-04	2.202306E-04
5.292000E-02	3.734903E-05	6.469076E-05
6.272000E-02	1.097070E-05	1.900192E-05
7.252000E-02	3.222820E-06	5.582117E-06
5.096000E-03	7.178532E-03	1.243365E-02
1.450400E-02	4.463186E-03	7.730505E-03
6.272000E-03	7.547924E-03	1.307346E-02
1.332800E-02	5.039151E-03	8.728112E-03
7.448000E-03	7.532047E-03	1.304596E-02
1.215200E-02	5.637561E-03	9.764594E-03
8.624000E-03	7.246194E-03	1.255084E-02
1.097600E-02	6.233134E-03	1.079616E-02
5.331200E-03	7.289991E-03	1.262670E-02
7.212800E-03	7.560291E-03	1.309488E-02
5.566400E-03	7.381641E-03	1.278545E-02
6.977600E-03	7.576902E-03	1.312365E-02
5.801600E-03	7.454637E-03	1.291188E-02
6.742400E-03	7.580778E-03	1.313036E-02
6.036800E-03	7.509650E-03	1.300716E-02
6.507200E-03	7.571350E-03	1.311403E-02
********************	PTV	PTV
6.742400E-03	7.580778E-03	1.313036E-02